IN THE CLAIMS:

The following claim listing replaces all previous claim listings. Please amend claims 1, 5, 6, 10, and 12, and cancel claim 4 as follows:

Claim 1. (original) A method for generating recognition models, the method comprising:

receiving a first model based on a first set of training data, the first set of training data originating from a first set of common entities;

receiving a second model based on a second set of training data, the second set of training data originating from a second set of common entities;

determining a difference in model information between the first model and the second model; and

creating an independent model based on the first set of training data and the second set of training data if the difference in model information is insignificant.

Claim 2. (original) The method of claim 1, wherein whether the model information is insignificant is based on a threshold model quantity.

Claim 3. (original) The method of claim 1, wherein determining the difference in model information includes calculating a Kullback Leibler distance between the first model and second model.

Claim 4. (original) The method of claim 3, wherein whether the model information is insignificant is based on a threshold Kullback Leibler distance quantity.

Claim 5. (original) The method of claim 1, wherein the first, second, and independent models are Gaussian mixture models.

Claim 6. (original) A system for generating recognition models, the method comprising:

a first model based on a first set of training data, the first set of training data originating from a first set of common entities;

a second model based on a second set of training data, the second set of training data originating from a second set of common entities; and a processing module configured to create an independent model based on the first set of training data and the second set of training data if the difference in model information between first model and the second model is insignificant.

Claim 7. (original) The system of claim 6, wherein whether the model information is insignificant is based on a threshold model quantity.

Claim 8. (original) The system of claim 6, wherein the processing model is further configured to calculate a Kullback Leibler distance between the first model and second model.

Claim 9. (original) The system of claim 8, wherein whether the model information is insignificant is based on a threshold Kullback Leibler distance quantity.

Claim 10. (original) The method of claim 6, wherein the first, second, and independent models are Gaussian mixture models.

Claim 11. (currently amended) A computer program product embodied in ~~a tangible media~~ computer memory comprising:

computer readable program codes coupled to the ~~tangible media~~ computer memory for generating recognition models, the computer readable program codes configured to cause the program to:

receive a first model based on a first set of training data, the first set of training data originating from a first set of common entities;

receive a second model based on a second set of training data, the second set of training data originating from a second set of common entities;

determine a difference in model information between the first model and the second model; and

create an independent model based on the first set of training data and the second set of training data if the difference in model information is insignificant.

Claim 12. (original) The computer program product of claim 11, wherein whether the model information is insignificant is based on a threshold model quantity.

Claim 13. (original) The computer program product of claim 11, wherein determining the difference in model information includes calculating a Kullback Leibler distance between the first model and second model.

Claim 14. (original) The computer program product of claim 13, wherein whether the model information is insignificant is based on a threshold Kullback Leibler distance quantity.

Claim 15. (original) The computer program product of claim 11, wherein the first, second, and independent models are Gaussian mixture models.

Claim 16. (original) A system for generating recognition models, the method comprising:

a first model based on a first set of training data, the first set of training data originating from a first set of common entities;

a second model based on a second set of training data, the second set of training data originating from a second set of common entities; and

means for creating an independent model based on the first set of training data and the second set of training data if the difference in model information between first model and the second model is insignificant.

Claim 17. (original) A method for recognizing data from a data stream originating from one of a plurality of data classes, the method comprising:

receiving a current feature vector;

computing a current vector probability that the current feature vector belongs to one of the plurality of data classes;

computing an accumulated confidence level that the data stream belongs to one of the plurality of data classes based on the current vector probability and on previous vector probabilities;

weighing class models based on the accumulated confidence; and

recognizing the current feature vector based on the weighted class models.

Claim 18. (original) The method of claim 17, wherein computing the current vector probability includes estimating an a posteriori class probability for the current feature vector.

Claim 19. (original) The method of claim 17, wherein computing the accumulated confidence level further comprising weighing the current vector probability more than the previous vector probabilities.

Claim 20. (original) The method of claim 17, further comprising determining if another feature vector is available for analysis.

Claim 21. (original) A system for recognizing data from a data stream originating from one of a plurality of data classes, the system comprising:

a receiving module configured to receive a current feature vector;

a first computing module configured to compute a current vector probability that the current feature vector belongs to one of the plurality of data classes;

a second computing module configured to compute an accumulated confidence level that the data stream belongs to one of the plurality of data classes based on the current vector probability and on previous vector probabilities;

a weighing module configured to weigh class models based on the accumulated confidence; and

a recognizing module configured to recognize the current feature vector based on the weighted class models.

Claim 22. (original) The system of claim 21, wherein the first computing module is further configured to estimate an a posteriori class probability for the current feature vector.

Claim 23. (original) The system of claim 21, wherein the second computing module is further configured to weigh the current vector probability more than the previous vector probabilities.

Claim 24. (currently amended) A computer program product embodied in ~~a tangible media~~ computer memory comprising:

computer readable program codes coupled to the ~~tangible media~~ computer memory for recognizing data from a data stream originating from one of a plurality of data classes, the computer readable program codes configured to cause the program to:

receive a current feature vector;

compute a current vector probability that the current feature vector belongs to one of the plurality of data classes;

compute an accumulated confidence level that the data stream belongs to one of the plurality of data classes based on the current vector probability and on previous vector probabilities;

weigh class models based on the accumulated confidence; and

recognize the current feature vector based on the weighted class models.

Claim 25. (original) The computer program product of claim 24, wherein the program code configured to compute the current vector probability includes program code configured to determine an a posteriori class probability for the current feature vector.

Claim 26. (original) The computer program product of claim 24, wherein the program code configured to compute the accumulated confidence level includes program code configured to weigh the current vector probability more than the previous vector probabilities.

Claim 27. (original) The computer program product of claim 24, further comprising program code configured to determine if another feature vector is available for analysis.